Mining High Utility Patterns in Large Databases using MapReduce Framework

[1]Ms. Priti Haribhau Deshmukh , [2]Assistant Prof. A. S. More

[1]Computer Engineering Department , Rajarshi Shahu School of Engineering and Research Narhe, Pune, 411041, India
[2]Computer Engineering Department Rajarshi Shahu School of Engineering and Research, Narhe, Pune, 411041, India

---***---

Abstract - *Recently considering users interest is more important and utility mining does not consider this interestingness measure. Utility mining is a new promising technology of data mining. High utility pattern growth approach is a look ahead strategy, and a linear data structure. Here linear data structure allows computing a tight bound for powerful pruning search space and to straightforwardly identify high utility patterns in an efficient and scalable way. In this it targets the root cause with prior algorithms. Recently, in data mining area high utility pattern mining is one of the most fundamental research issues since its ability to consider the no binary frequency values of items in transactions and different profit standards for every item. When a database is updated, or when the minimum threshold is changed, incremental and interactive data mining provides the ability of reducing unnecessary calculations using previous data structures and mining outcome of. In this analytical study, novel tree structures are proposed for efficient presentation of incremental and interactive HUP mining.*

Keyword —data mining, utility mining, high utility patterns , d2HUP, pattern mining.

1. INTRODUCTION

Frequent pattern mining is provided with the explanation that, candidate set generation and test prototype of prior. It is having many drawbacks like it requires multiple database scans and generates many candidate item sets. This query is solved by growth approach. For this introduction of a prefix-tree-based algorithm is provided that is without candidate set generation and testing. Given that frequent pattern mining plays an important role in data mining applications, its two limitations are found in this. First, it treats all items with the same importance/weight/price. Second one is, in one transaction, each item appears in a binary form, i.e., either it is present or absent. Since, in the real world, each item in the supermarket has a different importance/price and one customer can buy multiple copies of an item. Items that have high and low selling frequencies may have low and high profit values, respectively. Take example as, some frequently sold items like biscuits, milk, and pencil may have lower profit values compared to the infrequently sold higher profit value items such as gold ring and gold necklace. So, in finding only the traditional frequent patterns in a database cannot satisfy the requirement of finding the most valuable item sets/customers that contribute to the major part of the total profits in a retail business. This gives the motivation to develop a mining model to discover the item sets/customers contributing to the majority of the profit. Utility mining model was defined for discovering more important knowledge from a database. Here the importance of an item set by the concept of utility is measured. The dataset with no binary frequency standards of each item in transactions, and also with different profit standards of each item is handled. Therefore, utility mining represents real world market data. Other application areas, such as stock tickers, network traffic measurements, web server logs, data feeds from sensor networks, and telecom call records can have similar solutions. It is not suitable for large databases e.g. big data. Earlier studies deals with only small datasets, here big data is considered where data is having properties like variety , variability, veracity, volume, velocity, etc. Utility mining is done for big data which improves the efficiency.

2. REVIEW OF LITERATURE

As of now a great amount of study has been done on high utility pattern mining. In this section, we review the prior works on pattern mining and utility mining. In [1] Luc De Raedt implements the data mining using the constrained programming for item set mining. But this concept arise the problem of Specialization and optimization. An algorithm for finding patterns was presented by Ramesh C. Agarwal in [2] that finds long patterns using depth first search. This requires post-processing of data and time required for that depends on number of item sets found by tree generator.

In [3] Chowdhury Farhan Ahmed proposes three novel tree structures to efficiently perform incremental and interactive HUP mining. But this is not scalable for handling a large number of distinct items and transactions. Francesco Bonchi in [4] proposes a preprocessing approach overcomes the suspected incompatibility between anti-monotonicity and monotonicity, instead using the two components' synergy to reduce data and search-space size.In [5] Claudio Lucchese finds frequent patterns by using the concept of review and extends the state-of-the-art of the constraints that can be pushed in a frequent pattern computation. But this approach needs to build a system with incremental mining. Francesco Bonchi integrates the recently proposed ExAnte data reduction technique within the FP-growth algorithm in [6]. C.H. Cai introduced Mining Association Rules with Weighted Items in [7]. This approach introduces the notion of weighted items to represent the importance of individual items. But this approach considers mining of binary association rules. Roberto J. Bayardo proposed mining the most interesting rules in [8], to explain that the best rule according to any of these metrics must exist in along a support border. This approach suffers from deficiency of most optimized rule miners.

In [9] Rakesh Agrawal proposed fast algorithms for mining association rules. This approach present two, Apriority and Apriority Tied for discovering all significant association rules between items in a large database of transactions. It didn't consider the quantities of items brought in a transaction. Rakesh Agrawal proposed mining association rules between sets of items in large databases in [10]. This approach presented an efficient algorithm that generates all significant association rules between items in the database. But in this approach some parameters are redundant and are not generated in the calculations.

3. SYSTEM OVERVIEW

We can overcome the disadvantage of the existing method. In the existing system a single dataset is used for utility mining. But this is not suitable for large databases. So our proposed system works on big data using the parallel and distributed algorithms. Through which data can be partitioned and computed across many of hosts, and the execute application computations in parallel close to their data. Partitioning is made by use of Map Reduce framework of Hadoop. Each partition of data is individually mined using D2HUP algorithm and finally all results are collected together and set of high utility patterns are obtained.

A. Architecture

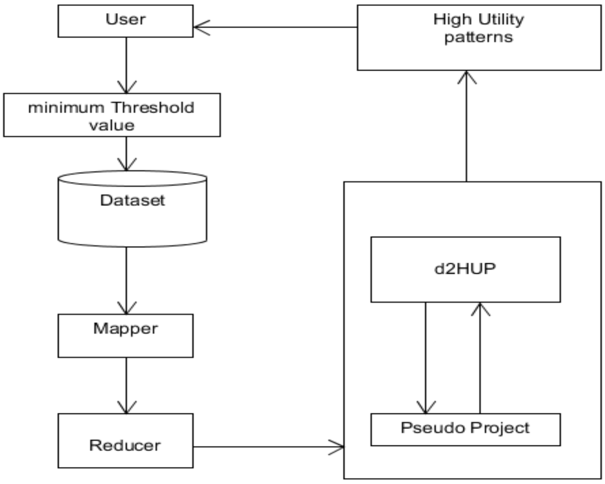

Figure 1. Architecture of proposed system

B. Module Description

1. Map Reduce module

Providing the dataset as the input of system, the system should partition data using Map-Reduce programming model. Mapper will partition data. For each partitioned data, high utility patterns are discovered using d2HUP algorithm.

2. Tree node edges finding Module

This module is used to generate a reverse set enumeration tree based on the product ID. This module will help us to generate the pattern of products.

3. Pattern Mining Module

This algorithm provides the whole implementation of Direct Discovery of High Utility Pattern. The algorithm calculates the result of each unit that was partitioned before individually.

4. Result Analysis

The result of above module is then passed to this module. This module takes the resulted utilization of each partition and compares it with the user defined threshold which will give us the final output and shows us the list of combination that has high utilization.

C. Mathematical Model

Let S be the High Utility Pattern system

S={ D, XUT, minU, TS, X, Ω, DFS, u, s, W, uBitem, uBfpe}

Where,

D-database

XUT- External Utility table

minU- minimum Threshold

TS- Transaction set
X- pattern
Ω- imposed ordering
DFS- Depth First Search
u- utilization value

$$u(X) = \sum_{t \in TS(X)} u(X,t) = \sum_{t \in TS(X)} \sum_{i \in X} u(i,t).$$

s- support value
W-relevant items

uBitem(i, X) - For a pattern X and an item (i U X), sum of utility of the full prefix extension of pattern with respect to every transaction in TS({i}U X)

$$uB_{item}(i,X) = \sum_{t \in TS(\{i\} \cup X)} u(fpe(X,t),t) \geq u(Y)$$

uBfpe - Sum of utility of the full prefix extension of pattern with respect to every transaction in TS(X)

$$uB_{fpe}(X) = \sum_{t \in TS(X)} u(fpe(X,t),t) \geq u(Y)$$

4. ALGORITHM

The proposed system uses following algorithms for finding the high utility pattern mining.

Algorithm 1: High Utility Pattern Algorithm

Input: minimum threshold and transaction dataset

Output: High Utility Patterns

```
Mapper<T, t1, t2...tn>
    T= {t1, t2 ...tn}
Reducer<t1, t2,...tn}
    If (T not Null) then
    For each T: t1, t2....tn then
    Item_info = tn (price, name_of_item)
    HUP = DFS (tn, threshold, Item_info)
    End for
End If
Else insert data first
Return HUP
```

Algorithm 2: Depth First Search Algorithm

DFS (*N*, *TS*(*pat*(*N*)), *minU*, Ω)

```
If u(pat(N)) ≥ minU then output pat(N)
W ← {i|i < pat(N)∧uBitem(i, pat(N))≥minU}
If Closure(pat(N),W, minU) is satisfied
then output nonempty subsets of W∪pat(N)
Else if Singleton(pat(N),W, minU) is satisfied
then output W ∪ pat(N) as an HUP
Else for each item i belongs to W in Ω do
If uBfpe({i} ∪ pat(N)) ≥ minU
then C ← the child node of N for i
TS(pat(C)) ← Project(TS(pat(N)), i)
DFS(C, TS(pat(C)), minU, Ω)
End for each
```

Algorithm 3: PseudoProject

PseudoProject (TScaul(pat(P)), i)

```
For each relevant item j < i do
(s[j], u[j],uBitem[j],uBfpe[j], link[j]) ← 0
End for each
For each utility list t threaded by link[i] do
u(pat(N), t) ← u(pat(P), t) + u(i, t)
Σ ← u(pat(N), t)
For each relevant item j belongs to t ∧ j < i by Ω do
s[j] ← s[j] + 1
u[j] ← u[j] + u(j, t) + u(pat(N), t)
Σ ← Σ + u(j, t)
uBfpe[j] ← uBfpe[j] + Σ
End for each
For each relevant item j belongs to t ∧ j < i by Ω do
uBitem[j] ← uBitem[j] + Σ
thread t into the chain by link[j]
End for each
End for each
```

The d2HUP algorithms traverse the reverse set enumerated tree using depth first search technique. The algorithm uses closure and singleton method which checks the utilization value of each node in tree. Using Pseudo projection algorithm it calculates utility value for each node in tree. With a parallel, distributed algorithm large data sets are processed. i.e a complex data structure which avoids costly repeated database scan. Divide and conquer methods can be used to decompose mining tasks into set of smaller tasks for mining.

5. SOFTWARE REQUIREMENT SPECIFICATION

Hardware Requirements

For this we required Pentium IV or latest system with hard disk of 40GB and 4GB RAM.

Software Requirements

Windows 8 or latest operating system is required for designed system. Java is the coding language is used. Back end is MS SQL Server. Eclipse IDE for Java is used as development environment. Hadoop 2.7.1 is also used.

6. PERFORMANCE ANALYSIS

The proposed system finds high utility patterns for large dataset. In this analytical study the items from foodmart dataset are calculated which are having more utility than threshold utility given by user. The final result provided by system is the patterns with high utilization according to user threshold. The proposed system is designed in such a way that it can provide facility of selecting dataset to users as per their requirement.

Performance is analyzed on the basis of execution time required. When we use existing algorithms to find high utility patterns from large datasets it takes long time. As compared to existing system this proposed system requires less time for execution.

In order to make efficient system we used Hadoop technology which partition the data using Map Reduce programming module and perform function on each partition in parallel manner which save system time. The proposed system will find result more accurately and with efficiently as compared to other system.

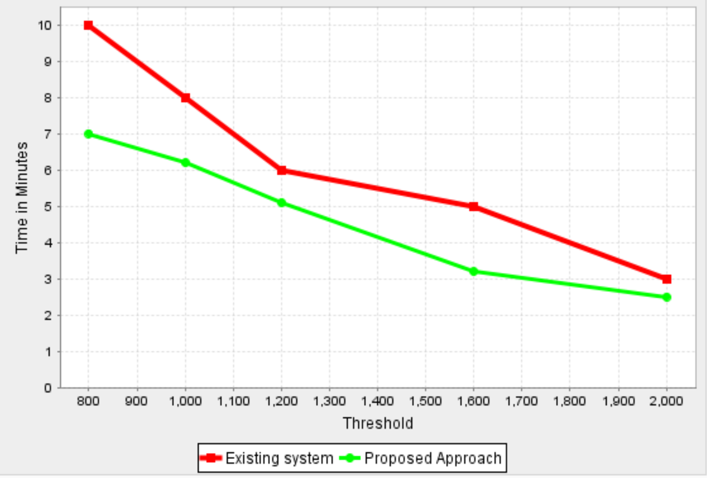

Chart.1 Comparison of execution time required by approach

7. CONCLUSION

The designed system implements d2HUP algorithm for utility mining with the item set share framework, which finds high utility patterns in big data without candidate generation.

Map Reduce framework of Hadoop is used for handling large databases. This approach is used to enhance the significance by the look ahead strategy that identifies high utility patterns.

ACKNOWLEDGEMENT

I express true sense of gratitude towards my project guide Prof. A. S. More of computer department for her invaluable co-operation and guidance that she gave me throughout my research, for inspiring me and providing me all the lab facilities, which made this research work very convenient and easy. I would also like to express my appreciation and thanks to our HOD Prof. R. H. Kulkarni and Director Dr A. B. Auti and all my friends who knowingly or unknowingly have assisted me throughout my hard work.

REFERENCES

[1] L. De Raedt, T. Guns, and S. Nijssen, “Constraint programming for item set mining,” in SIGKDD, pp. 204–212, 2008.

[2] R. Agarwal, C. Aggarwal, and V. Prasad, “Depth first generation of long patterns,” in SIGKDD, pp. 108–118, 2000.

[3] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, “Efficient tree structures for high utility pattern mining in

incremental databases," IEEE TKDE, vol. 21, no. 12, pp. 1708– 1721, 2009.

[4] F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, "Exante: A preprocessing method for frequent-pattern mining," IEEE Intelligent Systems, vol. 20, no. 3, pp. 25–31, 2005.

[5] F. Bonchi and C. Lucchese, "Extending the state-of-the-art of constraint-based pattern discovery," Data and Knowledge Engineering, vol. 60, no. 2, pp. 377–399, 2007.

[6] F. Bonchi and B. Goethals, "Fp-bonsai: The art of growing and pruning small fp-trees," in PAKDD, pp. 155–160, 2004.

[7] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong, "Mining association rules with weighted items," in International Database Engineering and Applications Symposium. IEEE, 68-77, 1998.

[8] R. Bayardo and R. Agrawal, "Mining the most interesting rules," in SIGKDD. ACM, pp. 145–154, 1999.

[9] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in VLDB, pp. 487–499, 1994.

[10] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in SIGMOD. ACM, pp. 207–216, 1993.

[11] S. Dawar and V. Goyal, "Up-hist tree: An efficient data structure for mining high utility patterns from transaction databases," in IDEAS. ACM, pp. 56–61, 2015.

[12] T. De Bie, "Maximum entropy models and subjective interestingness: an application to tiles in binary databases," Data Mining and Knowledge Discovery, vol. 23, no. 3, pp. 407–446,2011.

[13] G.-C. Lan, T.-P. Hong, and V. S. Tseng, "An efficient projectionbased indexing approach for mining high utility itemsets," KAIS, vol. 38, no. 1, pp. 85–107, 2014.

[14] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, "Isolated items discarding strategy for discovering high utility itemsets," Data & Knowledge Engineering, vol. 64, no. 1, pp. 198–217, 2008.

[15] A. Erwin, R. P. Gopalan, and N. R. Achuthan, "Efficient mining of high utility itemsets from large datasets," in PAKDD,

pp. 554–561, 2008.

[16] P. Fournier-Viger, C.-W. Wu, S. Zida, and V. S. Tseng, "Fhm: Faster high-utility itemset mining using estimated utility cooccurrence pruning," in ISMIS. Springer, pp. 83–92, 2014.

[17] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey," ACM Computing Surveys, vol. 38, no. 3, p. 9, 2006.

[18] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in SIGMOD. ACM, pp. 1–12, 2000.

[19] R. J. Hilderman, C. L. Carter, H. J. Hamilton, and N. Cercone, "Mining market basket data using share measures and characterized itemsets," in PAKDD, pp. 72–86, 1998.

[20] R. J. Hilderman and H. J. Hamilton, "Measuring the interestingness of discovered knowledge: A principled approach," Intelligent Data Analysis, vol. 7, no. 4, pp. 347–382, 2003.